

### **Amendments to the Claims**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (currently amended) A method for finding a local extrema for a processing element having a set of values associated therewith, the method comprising:
  - separating said set of values into an odd ~~numbered~~ set corresponding to values in odd positions within said set and an even ~~numbered~~ set corresponding to values in even positions within said set;
  - determining ~~an odd~~ a first extrema from said odd ~~numbered~~ set;
  - determining ~~an even~~ a second extrema from said even ~~numbered~~ set; and
  - determining said local extrema from said ~~odd~~ first extrema and said ~~even~~ second extrema;
  - and
  - storing said local extrema.
2. (currently amended) The method of claim 1 wherein said separating said set of values into an odd ~~numbered~~ set and an even ~~numbered~~ set comprises:
  - loading a value from an odd position within said set into a first register;
  - loading a value from an even position within said set into a second register and transferring said value in said first register to a third register;
  - loading a value from a next odd position within said set into said first register and transferring said value in said second register to a fourth register; and
  - loading a value from a next even position within said set into said second register.
3. (currently amended) The method of claim 2 wherein said determining ~~an odd~~ a first extrema from said odd ~~numbered~~ set comprises comparing the value in said first register to the value in said third register.
4. (currently amended) The method of claim ~~[[1]]~~ 3 wherein said determining further comprises:
  - selecting the greater value from said first register and said third register if a high ~~odd~~ first extrema is desired; and

selecting the lesser value from said first register and said third register if a low ~~odd~~ first extrema is desired.

5. (currently amended) The method of claim ~~[[1]]~~ 2 wherein said determining ~~an even~~ a second extrema from said even set of values comprises comparing the value in said second register to the value in said fourth register.

6. (currently amended) The method of claim 5 wherein said determining further comprises:  
selecting the greater value from said second register and said fourth register if a high ~~even~~ second extrema is desired; and  
selecting the lesser value from said second register and said fourth register if a low ~~even~~ second extrema is desired.

7. (currently amended) The method of claim 1 wherein said determining said local extrema from said ~~odd~~ first extrema and said ~~even~~ second extrema further comprises:  
selecting the greater value from said ~~odd~~ first extrema and said ~~even~~ second extrema if a local high extrema is desired; and  
selecting the lesser value from said ~~odd~~ first extrema and said ~~even~~ second extrema if ~~the a~~ a local low extrema is desired.

8. (currently amended) The method of claim ~~[[1]]~~ 2 further comprising:  
storing said ~~odd~~ first extrema in said third register;  
loading another value from an odd position within said set into said first register;  
comparing the value in said first register to the value in said third register; and  
repeating said storing, loading and comparing steps for remaining values within an odd position within said set.

9. (currently amended) The method of claim ~~[[5]]~~ 2 further comprising:  
storing said ~~even~~ second extrema in said fourth register;  
loading another value from an even position within said set into said second register;

comparing the value in said second register to the value in said fourth register; and  
repeating said storing, loading and comparing steps for remaining values within an even position within said set.

10. (currently amended) The method of claim 1 wherein said separating said set of values into an odd ~~numbered~~ set and an even ~~numbered~~ set comprises:

loading a first portion of an odd numbered value within said set into a first register;  
transferring said first portion of said odd numbered value from said first register into a second register and loading a second portion of said odd numbered value into said first register;  
transferring said second portion of said odd numbered value from said first register into a third register and loading a first portion of an even numbered value from within said set into said first register;  
transferring said first portion of said even numbered value from said first register into a fourth register and loading a second portion of said even numbered value into said first register;  
transferring said second portion of said even numbered value from said first register into a fifth register.

11. (currently amended) The method of claim 1 wherein said separating said set of values into an odd ~~numbered~~ set and an even ~~numbered~~ set comprises:

loading said least significant byte of a short value from an odd position within said set into a first register;  
transferring said least significant byte of a short from an odd position in said first register to a second register and loading said most significant byte of said short value from said odd position within said set into said first register;  
transferring said most significant byte of a short from an odd position in said first register to a third register, loading said least significant byte of a short value from an even position within said set into said first register, and initializing a fourth register with said least significant byte of a short from an odd position in said second register;  
transferring said least significant byte of a short value from an even position within said set from said first register to a sixth register, loading said most significant byte of said short value from said even position within said set into said first register, and initializing a fifth register with said most significant byte of a short from said odd position in said third register;

transferring said most significant byte of said short value from said even position within said set from said first register to a seventh register, loading a least significant byte of a short value from another odd position within said set into said first register, and initializing ~~[[a]]~~ an eighth register with said least significant byte of a short from said even position in said sixth register;

transferring said least significant byte of said short value from said another odd position within said set from said first register to said second register, loading a most significant byte of said short value from said another odd position within said set into said first register, and initializing a ninth register with said most significant byte of a short from said even position in said seventh register;

transferring said most significant byte of said short value from said another odd position within said set from said first register to said third register, and loading a least significant byte of a short value from another even position within said set into said first register;

transferring said least significant byte of said short value from said another even position within said set from said first register to said sixth register, and loading a most significant byte of said short value from said another even position within said set into said first register; and

transferring said most significant byte of said short value from said another even position within said set from said first register to said seventh register.

12. (currently amended) The method of claim ~~44~~ 1 wherein said determining ~~an odd~~ a first extrema from said odd ~~numbered~~ set comprises comparing ~~said a~~ said a least significant byte of ~~said a~~ said a short value from said odd position within said set to ~~said a~~ said a least significant byte of ~~said a~~ said a short value from said another odd position within said set and comparing ~~said a~~ said a most significant byte of said short value from said odd position within said set to ~~said a~~ said a most significant byte of said short value from said another odd position within said set.

13. (currently amended) The method of claim ~~44~~ 1 wherein said determining ~~an even~~ a second extrema from said even ~~numbered~~ set comprises comparing ~~said a~~ said a least significant byte of ~~said a~~ said a short value from said even position within said set to ~~said a~~ said a least significant byte of ~~said a~~ said a short value from said another even position within said set and comparing ~~said a~~ said a most significant byte of said short value from said even position within said set to ~~said a~~ said a most significant byte of said short value from said another even position within said set.

14. (currently amended) A method comprising:

identifying bytes of data within a data stream as having one of an odd or an even position;  
processing said bytes of data having an odd position to produce ~~an odd~~ a first extrema;  
processing said bytes of data having an even position to produce ~~an even~~ a second extrema;  
~~and~~

determining a local extrema from said ~~odd~~ first extrema and said ~~even~~ second extrema; and  
storing said local extrema.

15. (currently amended) The method of claim 14 wherein said processing said bytes of data having an odd position and processing said bytes of data having an even position comprises:

loading a byte of data having an odd position into a first register;

loading a byte of data having an even position into a second register and transferring said byte of data in said first register into a third register;

loading a byte from a next odd position within said data stream into said first register and transferring said byte of data in said second register into a fourth register;

comparing said byte of data in said first register to said byte of data in said third register to produce said ~~odd~~ first extrema and loading a byte of data from a next even position within said data stream into said second register; and

comparing said byte of data in said second register to said byte of data in said fourth register to produce said ~~even~~ second extrema.

16. (currently amended) The method of claim 15 wherein said processing said bytes of data having an odd position to produce ~~an odd~~ said first extrema further comprises:

selecting the greater valued byte from said first register and said third register if a high ~~odd~~ first extrema is desired; and

selecting the lesser valued byte from said first register and said third register if a low ~~odd~~ first extrema is desired.

17. (currently amended) The method of claim 15 wherein said processing said bytes of data having an even position to produce ~~an even~~ said second extrema further comprises:

selecting the greater valued byte from said second register and said fourth register if a high ~~even~~ second extrema is desired; and

selecting the lesser valued byte from said second register and said fourth register if a low ~~even~~ second extrema is desired.

18. (currently amended) The method of claim 15 wherein said determining a local extrema from said ~~odd~~ first extrema and said ~~even~~ second extrema further comprises:

selecting the greater valued byte from said ~~odd~~ first extrema and said ~~even~~ second extrema if a local high extrema is desired; and

selecting the lesser valued byte from said ~~odd~~ first extrema and said ~~even~~ second extrema if the local low extrema is desired.

19. (currently amended) The method of claim 15 further comprising:

storing said ~~odd~~ first extrema in said third register;

loading another byte from an odd position within said data stream into said first register;

comparing the byte within said first register to the byte within said third register; and

repeating said storing, loading and comparing steps for remaining bytes within an odd position within said data stream.

20. (currently amended) The method of claim 15 further comprising:

storing said ~~even~~ second extrema in said fourth register;

loading another byte from an even position within said data stream into said second register;

comparing the byte within said second register to the byte within said fourth register; and

repeating said storing, loading and comparing steps for remaining bytes within an even position within said data stream.

21. (currently amended) A method for determining a local extrema for a processing element, comprising:

loading odd numbered bytes of data into a first plurality of registers;

loading even numbered bytes of data into a second plurality of registers;

comparing certain of said loaded odd numbered bytes to produce ~~an odd~~ a first extrema;

comparing certain of said loaded even numbered bytes to produce ~~an even~~ a second extrema;

and

producing a local extrema in response to said ~~odd~~ first extrema and said ~~even~~ second extrema;  
and  
storing said local extrema.

22. (original) The method of claim 21 wherein said loading odd numbered bytes of data into a first plurality of registers comprises:

loading a byte of data having an odd position into a first register;  
transferring said byte of data in said first register into a third register; and  
loading a byte from a next odd position within said data stream into said first register.

23. (original) The method of claim 21 wherein said loading even numbered bytes of data into a second plurality of registers comprises:

loading a byte of data having an even position into a second register;  
transferring said byte of data in said second register into a fourth register; and  
loading a byte of data from a next even position within said data stream into said second register.

24. (currently amended) The method of claim 22 wherein said comparing certain of said loaded odd numbered bytes to produce ~~an odd~~ said first extrema comprises comparing said byte of data in said first register to said byte of data in said third register to produce said ~~odd~~ first extrema.

25. (currently amended) The method of claim 23 wherein said comparing certain of said loaded even numbered bytes to produce ~~an even~~ said second extrema comprises comparing said byte of data in said second register to said byte of data in said fourth register to produce said ~~even~~ second extrema.

26. (currently amended) The method of claim 24 wherein said comparing certain of said loaded odd numbered bytes to produce ~~an odd~~ said first extrema further comprises:

storing said ~~odd~~ first extrema in said third register;  
loading another byte from an odd position within said data stream into said first register;  
comparing the byte within said first register to the byte within said third register; and  
repeating said storing, loading and comparing steps for remaining bytes within an odd position within said data stream.

27. (currently amended) The method of claim 25 wherein said comparing certain of said loaded even numbered bytes to produce ~~an even~~ said second extrema further comprises:

storing said ~~even~~ second extrema in said fourth register;  
loading another byte from an even position within said data stream into said second register;  
comparing the byte within said second register to the byte within said fourth register; and  
repeating said storing, loading and comparing steps for remaining bytes within an even position within said data stream.

28. (currently amended) The method of claim 14 wherein said processing said bytes of data having an odd position and processing said bytes of data having an even position comprises:

loading said least significant byte of a short value from an odd position within said set into a first register;

transferring said least significant byte of a short from an odd position in said first register to a second register and loading said most significant byte of said short value from said odd position within said set into said first register;

transferring said most significant byte of a short from an odd position in said first register to a third register, loading said least significant byte of a short value from an even position within said set into said first register, and initializing a fourth register with said least significant byte of a short from an odd position in said second register;

transferring said least significant byte of a short value from an even position within said set from said first register to a sixth register, loading said most significant byte of said short value from said even position within said set into said first register, and initializing a fifth register with said most significant byte of a short from said odd position in said third register;

transferring said most significant byte of said short value from said even position within said set from said first register to a seventh register, loading a least significant byte of a short value from another odd position within said set into said first register, and initializing ~~[[a]]~~ an eighth register with said least significant byte of a short from said even position in said sixth register;

transferring said least significant byte of said short value from said another odd position within said set from said first register to said second register, loading a most significant byte of said short value from said another odd position within said set into said first register, and



initializing a ninth register with said most significant byte of a short from said even position in said seventh register;

transferring said most significant byte of said short value from said another odd position within said set from said first register to said third register, and loading a least significant byte of a short value from another even position within said set into said first register;

transferring said least significant byte of said short value from said another even position within said set from said first register to said sixth register, and loading a most significant byte of said short value from said another even position within said set into said first register; and

transferring said most significant byte of said short value from said another even position within said set from said first register to said seventh register.

29. (currently amended) The method of claim ~~28~~ 14 wherein said determining ~~an odd~~ said first extrema from said odd ~~numbered~~ set comprises comparing said a least significant byte of said a short value from said odd position within said set to said a least significant byte of said a short value from said another odd position within said set and comparing said a most significant byte of said short value from said odd position within said set to said a most significant byte of said short value from said another odd position within said set.

30. (currently amended) The method of claim ~~28~~ 14 wherein said determining ~~an even~~ said second extrema from said even ~~numbered~~ set comprises comparing said a least significant byte of said a short value from said even position within said set to said a least significant byte of said a short value from said another even position within said set and comparing said a most significant byte of said short value from said even position within said set to said a most significant byte of said short value from said another even position within said set.

31. (currently amended) A computer readable memory device carrying a set of instructions which, when executed, perform a method comprising:

separating said set of values into an odd ~~numbered~~ set and an even ~~numbered~~ set;

determining ~~an odd~~ a first extrema from said odd ~~numbered~~ set;

determining ~~an even~~ a second extrema from said even ~~numbered~~ set; and

determining said local extrema from said ~~odd~~ first extrema and said ~~even~~ second extrema;

and

Appl. No.: 10/689,335

Docket No.: PAT001076-000

Response to Amdt. Dated: 20 November 2006

storing said local extrema.